

Connected Device Studio

Restful API Specification for Device

Monday, 8th Jan 2018

Version 1.0

Document Particulars

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Author | Date | Notes |
| 0.2 | Kevin Kao | 2018/01/02 | Initial Version |
| 1.0 | Kevin Kao | 2018/01/08 | First Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Document Circulation

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Email | Role | Company |
| Kevin Kao | kevika@microsoft.com | Senior Program Manager | Microsoft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Contents

Contents

[Document Particulars 2](#_Toc503184194)

[Contents 3](#_Toc503184195)

[Notes for API Implementation 1](#_Toc503184196)

[1 Environment and Awareness 1](#_Toc503184197)

[1.1 Restful API Service URI 1](#_Toc503184198)

[1.2 Get Authentication Token before call any other API 1](#_Toc503184199)

[1.3 Use access\_token for all following APIs Call 2](#_Toc503184200)

[2 Device Activation Endpoint 3](#_Toc503184201)

[2.1 Activate Device by IoTHubDeviceId and Activation Code 3](#_Toc503184202)

[3 Device Endpoint 4](#_Toc503184203)

[3.1 Get Device Detail 4](#_Toc503184204)

[3.2 Upload Device Log 4](#_Toc503184205)

[3.3 Get Message Schema associated with IoTDevice 4](#_Toc503184206)

[3.4 Update IoTHubDevice 4](#_Toc503184210)

[4 DeviceType Endpoint 5](#_Toc503184214)

[4.1 Get All DeviceType 5](#_Toc503184215)

[5 EquipmentClass Endpoint 6](#_Toc503184216)

[5.1 Get All EquipmentClass 6](#_Toc503184217)

[5.2 Get EquipmentClass by ID 6](#_Toc503184218)

[5.3 Add an EquipmentClass 6](#_Toc503184221)

[5.4 Delete EquipmentClass by ID 6](#_Toc503184222)

[5.5 Update EquipmentClass by ID 6](#_Toc503184223)

[5.6 Assign MessageCatalog for EquipmentClass by ID 7](#_Toc503184229)

[6 MessageCatalog Endpoint 8](#_Toc503184230)

[6.1 Get All MessageCatalog 8](#_Toc503184231)

[6.2 Get MessageCatalog with Elements by Id 8](#_Toc503184232)

[6.3 Add a MessageCatalog 8](#_Toc503184233)

[6.4 Delete a MessageCatalog 8](#_Toc503184234)

[6.5 Update a MessageCatalog by Id 8](#_Toc503184235)

[6.6 Create/Update/Delete MessageElement by MessageCatalog Id 9](#_Toc503184236)

[7 Equipment Endpoint 10](#_Toc503184237)

[7.1 Get All Equipment under IoTDevice 10](#_Toc503184238)

[7.2 Add an Equipment under IoTDevice 10](#_Toc503184239)

[7.3 Delete an Equipment under IoTDevice 10](#_Toc503184240)

[7.4 Update an Equipment under IoTDevice 10](#_Toc503184241)

[7.5 Upload Equipment Photo under IoTDevice 10](#_Toc503184242)

[8 IoT Hub Receiver Endpoint 12](#_Toc503184243)

[8.1 Restart IoT Hub Receiver 12](#_Toc503184244)

[Appendix 1: Object Definition 13](#_Toc503184245)

[IoT Device Object 13](#_Toc503184246)

[DeviceType Object 13](#_Toc503184247)

[MessageSchema Object 13](#_Toc503184248)

[EquipmentClass Object 13](#_Toc503184249)

[MessageCatalog Object 13](#_Toc503184250)

[MessageElement Object 14](#_Toc503184251)

[Equipment Object 14](#_Toc503184252)

[Appendix 2: Data Dictionary 15](#_Toc503184253)

[Appendix 3: Fail Responses 15](#_Toc503184254)

# Notes for API Implementation

1. IoTHubDeviceId + CompanyId are unique. To obtain token, client has to send over CompanyId and IoTHubDeviceId along with IoTHubDevicePW.
2. Store Id (integer, table key) of the IoTHubDevice on token. Using this Id (integer, table key) to access all other data.
3. Client doesn’t require to pass IoTHubDeviceId at all endpoint calls. Controller just retrieve Id (integer, table key) from token.
4. Client doesn’t require to pass any data that knew on API level:
   1. CompanyID
   2. FactoryID
   3. IoTHubDeviceID
5. Allowed permission. Device only allow to manipulate its’ own entities, such as update its’ meta-data, add/update/delete equipment under itself.
6. Device API doesn’t support child message management.

# Environment and Awareness

## Restful API Service URI

<https://msfapiservice.trafficmanager.net/>

The API web site used self-signed SSL certificate. Your application may need to handle by accepted self-signed certificate. You also can replace https with http by in-secure access protocol, but this is not recommended.

## Get Authentication Token before call any other API

Endpoint: /token

HTTP Encrypted: x-www-form-urlencoded

Form input

* grant\_type: password
* companyId: <CompanyId>
* email: <IoTHubDeviceId>
* password: <IoTHubDevicePW>
* role: device

***Successful Response:***

|  |  |
| --- | --- |
| HTTP Status Code | 200 OK |
| HTTP Result | {  "access\_token": "xxxxxxx",  "token\_type": "bearer",  "expires\_in": 7199,  ".issued": "Thu, 02 Nov 2017 02:30:55 GMT",  ".expires": "Thu, 02 Nov 2017 04:30:55 GMT"  } |

***Fail Response:***

|  |  |
| --- | --- |
| HTTP Status Code | 400 Bad Request |
| HTTP Result | {  "error": "Authentication Fail",  "error\_description": "Authentication Fail."  } |

Your application shall store *access\_token* at somewhere, and re-use it for all following APIs before it gets expired.

## Use access\_token for all following APIs Call

Endpoint: <Refer to Chapter 3 ~ 5 >

HTTP Encrypted: x-www-form-urlencoded

HTTP Header

* Authorization: bearer <access\_token> # bearer + space + access\_token

***Successful Response:***

|  |  |
| --- | --- |
| HTTP Status Code | 200 OK |
| HTTP Result | {  “Name”:”value”,  “Add”:”value”  } |

# Device Activation Endpoint

## Activate Device by IoTHubDeviceId and Activation Code

|  |  |
| --- | --- |
| Endpoint | /device-api/Device/Activate |
| Method | POST |
| QueryString |  |
| Allow Encrypted | x-www-form-urlencoded |
| POST Content | IoTHubDeviceId=xxxxxx&ActivationCode=xxxxxx |
| HTTP Response Status | 200 OK |
| Response Content | {  IoTDevice View Object  } |

\* This Endpoint doesn’t require <access\_token>

# Device Endpoint

## Get Device Detail

|  |  |
| --- | --- |
| Endpoint | /device-api/Device |
| Method | GET |
| QueryString |  |
| HTTP Response Status | 200 OK |
| Response Content | {  IoTDevice View Object  } |

## Upload Device Log

|  |  |
| --- | --- |
| Endpoint | /device-api/Device/Log |
| Method | POST |
| QueryString |  |
| Allow Encrypted | Multipart Content |
| POST Content | Attached File |
| HTTP Response Status | 200 OK |
| Response Content |  |

## Get Message Schema associated with IoTDevice

|  |  |
| --- | --- |
| Endpoint | /device-api/Device/MessageSchema |
| Method | GET |
| QueryString |  |
| HTTP Response Status | 200 OK |
| Response Content | [  { MessageSchema View Object }  ] |



## Update IoTHubDevice

|  |  |
| --- | --- |
| Endpoint | /device-api/Device |
| Method | PUT |
| QueryString |  |
| PUT Content | IoTDevice Update Object |
| HTTP Response Status | 200 OK |
| Response Content | {  IoTDevice View Object  } |



# DeviceType Endpoint

## Get All DeviceType

|  |  |
| --- | --- |
| Endpoint | /device-api/DeviceType |
| Method | GET |
| QueryString |  |
| HTTP Response Status | 200 OK |
| Response Content | [  { DeviceType View Object }  ] |

# EquipmentClass Endpoint

## Get All EquipmentClass

|  |  |
| --- | --- |
| Endpoint | /device-api/EquipmentClass |
| Method | GET |
| QueryString |  |
| HTTP Response Status | 200 OK |
| Response Content | [  {  EquipmentClass View Object  }  ] |

## Get EquipmentClass by ID

|  |  |
| --- | --- |
| Endpoint | /device-api/EquipmentClass/{Id} |
| Method | GET |
| QueryString |  |
| HTTP Response Status | 200 OK |
| Response Content | {  EquipmentClass View Object  } |



## Add an EquipmentClass

|  |  |
| --- | --- |
| Endpoint | /device-api/EquipmentClass |
| Method | POST |
| QueryString |  |
| POST Content | EquipmentClass Add Object |
| HTTP Response Status | 200 OK |
| Response Content | {  EquipmentClass View Object  } |

## Delete EquipmentClass by ID

|  |  |
| --- | --- |
| Endpoint | /device-api/EquipmentClass/{Id} |
| Method | DELETE |
| QueryString |  |
| HTTP Response Status | 200 OK |
| Response Content |  |

## Update EquipmentClass by ID

|  |  |
| --- | --- |
| Endpoint | /device-api/EquipmentClass/{Id} |
| Method | PUT |
| QueryString |  |
| PUT Content | EquipmentClass Update Object |
| HTTP Response Status | 200 OK |
| Response Content | {  EquipmentClass View Object  } |



## Assign MessageCatalog for EquipmentClass by ID

|  |  |
| --- | --- |
| Endpoint | /device-api/EquipmentClass/{Id}/MessageCatalog |
| Method | PUT |
| QueryString | rs=1 # Restart IoT Hub Receiver |
| PUT Content | Array of MessageCatalogId |
| Example | MessageCatalogId[0]=8&MessageCatalogId[1]=18&MessageCatalogId[2]=28 |
| HTTP Response Status | 200 OK |
| Response Content | [  { MessageCatalog View Object }  ] |

# MessageCatalog Endpoint

## Get All MessageCatalog

|  |  |
| --- | --- |
| Endpoint | /device-api/MessageCatalog |
| Method | GET |
| QueryString | cm=0 #Including child message. 0:false, 1:true |
| HTTP Response Status | 200 OK |
| Response Content | [  {  MessageCatalog View Object  }  ] |

## Get MessageCatalog with Elements by Id

|  |  |
| --- | --- |
| Endpoint | /device-api/MessageCatalog/{Id}/Element |
| Method | GET |
| QueryString |  |
| HTTP Response Status | 200 OK |
| Response Content | {  MessageCatalog View Object  [  { MessageElement View Object }  ]  } |

## Add a MessageCatalog

|  |  |
| --- | --- |
| Endpoint | /device-api/MessageCatalog |
| Method | POST |
| QueryString |  |
| POST Content | MessageCatalog Add Object |
| HTTP Response Status | 200 OK |
| Response Content | {  MessageCatalog View Object  } |

## Delete a MessageCatalog

|  |  |
| --- | --- |
| Endpoint | /device-api/MessageCatalog/{Id} |
| Method | DELETE |
| QueryString | rs=1 # Restart IoT Hub Receiver |
| HTTP Response Status | 200 OK |
| Response Content |  |

## Update a MessageCatalog by Id

|  |  |
| --- | --- |
| Endpoint | /device-api/MessageCatalog/{Id} |
| Method | PUT |
| QueryString | rs=1 # Restart IoT Hub Receiver |
| PUT Content | MessageCatalog Update Object |
| HTTP Response Status | 200 OK |
| Response Content | {  MessageCatalog View Object  } |

## Create/Update/Delete MessageElement by MessageCatalog Id

|  |  |
| --- | --- |
| Endpoint | /device-api/MessageCatalog/{Id}/MessageElement |
| Method | PUT |
| QueryString | rs=1 # Restart IoT Hub Receiver |
| PUT Content | Array of MessageElement Object |
| Example | Act[0]=c&Id[0]=0&ElementName[0]=name1&ElementDataType[0]=String&MandatoryFlag[0]=1&  Act[1]=u&Id[1]=12& ElementName[1]=name2&ElementDataType[1]=String&MandatoryFlag[1]=1&  Act[2]=d&Id[2]=20& ElementName[2]=name2&ElementDataType[2]=String&MandatoryFlag[2]=0 |
| Remark | Create Id[0]  Update Id[1]  Delete Id[2] |
| HTTP Response Status | 200 OK |
| Response Content | [  { MessageElement View Object }  ] |

# Equipment Endpoint

## Get All Equipment under IoTDevice

|  |  |
| --- | --- |
| Endpoint | /device-api/Device/Equipment |
| Method | GET |
| QueryString |  |
| HTTP Response Status | 200 OK |
| Response Content | [  {  Equipment View Object  }  ] |

## Add an Equipment under IoTDevice

|  |  |
| --- | --- |
| Endpoint | /device-api/Device/Equipment |
| Method | POST |
| QueryString | rs=1 # Restart IoT Hub Receiver |
| POST Content | Equipment New Object |
| HTTP Response Status | 200 OK |
| Response Content | {  Equipment View Object  } |

## Delete an Equipment under IoTDevice

|  |  |
| --- | --- |
| Endpoint | /device-api/Device/Equipment/{Id} |
| Method | Delete |
| QueryString | rs=1 # Restart IoT Hub Receiver |
| HTTP Response Status | 200 OK |
| Response Content |  |

## Update an Equipment under IoTDevice

|  |  |
| --- | --- |
| Endpoint | /device-api/Device/Equipment/{Id} |
| Method | PUT |
| QueryString | rs=1 # Restart IoT Hub Receiver |
| PUT Content | Equipment Update Object |
| HTTP Response Status | 200 OK |
| Response Content | {  Equipment View Object  } |

## Upload Equipment Photo under IoTDevice

|  |  |
| --- | --- |
| Endpoint | /device-api/Device/Equipment/{Id}/Photo |
| Method | POST |
| QueryString |  |
| Allow Encrypted | Multipart Content |
| POST Content | Attached File |
| HTTP Response Status | 200 OK |
| Response Content | {  Equipment View Object  } |

# IoT Hub Receiver Endpoint

## Restart IoT Hub Receiver

|  |  |
| --- | --- |
| Endpoint | /device-api/IoTHubReceiver/Restart |
| Method | PUT |
| QueryString |  |
| PUT Content | rs=1 # Restart IoT Hub Receiver |
| HTTP Response Status | 200 OK |
| Response Content |  |

# Appendix 1: Object Definition

## IoT Device Object

|  |  |
| --- | --- |
| Update | IoTHubProtocol=xxx  DeviceVendor=xxx  DeviceModel=xxx |
| View | {  “IoTHubDeviceID”:”xxx”,  “AuthenticationType”:”xxx”,  “CertFile”:”xxx”,  “KeyFile”:”xxx”,  “CertPW”:”xxx”,  “IoTHubDeviceKey”:”xxx”,  “IoTHubProtocol”:”xxx”,  “CompanyID”: 12,  “DeviceTypeId”: 1,  “DeviceVendor”:”xxx”,  “DeviceModel”:”xxx”  } |

## DeviceType Object

|  |  |
| --- | --- |
| View | {  “Id”:12,  “Name”:”xxx”,  “Description”:”xxx”  } |

## MessageSchema Object

|  |  |
| --- | --- |
| View | {  MessageCatalog View Object,  [  { MessageElement View Object }  ]  } |

## EquipmentClass Object

|  |  |
| --- | --- |
| Create | Name=xxx  Description=xxx |
| Update | Name=xxx  Description=xxx |
| View | {  “Id”:12,  “Name”:”xxx”,  “Description”:”xxx”  } |

## MessageCatalog Object

|  |  |
| --- | --- |
| Create | Name=xxx  Description=xxx  MonitorFrequenceInMinSec=1000 |
| Update | Name=xxx  Description=xxx  MonitorFrequenceInMinSec=1000 |
| View | {  “Id”:12,  “Name”:”xxx”,  “Description”:”xxx”,  “MonitorFrequenceInMinSec”:1000,  “ChildMessageFlag”:false  } |

## MessageElement Object

|  |  |
| --- | --- |
| Create | Act[0]=c&Id[0]=0&ElementName[0]=name1&ElementDataType[0]=String&MandatoryFlag[0]=1 |
| Update | Act[1]=u&Id[1]=12& ElementName[1]=name2&ElementDataType[1]=String&MandatoryFlag[1]=1 |
| Delete | Act[2]=d&Id[2]=20& ElementName[2]=name2&ElementDataType[2]=String&MandatoryFlag[2]=0 |
| View | {  “Id”:12,  “ElementName”:”xxx”,  “ElementDataType”:”xxx”,  “MandatoryFlag”:true  } |

## Equipment Object

|  |  |
| --- | --- |
| Create | EquipmentId=xxx  Name=xxx  EquipmentClassId=12  Latitude:23.999  Longitude:122.3333  MaxIdleInSec=30 |
| Update | EquipmentId=xxx  Name=xxx  EquipmentClassId=12  Latitude:23.999  Longitude:122.3333  MaxIdleInSec=30 |
| View | {  “Id”:12,  “EquipmentId”:”xxx”,  “Name”:”xxx”,  “EquipmentClassId”:10,  “Latitude”:25.11221,  “Longitude”:120.499809,  “MaxIdleInSec”:30,  “PhotoURL”:”xxx”  } |

# Appendix 2: Data Dictionary

|  |  |
| --- | --- |
| ElementDataType | datetime  string  numeric  bool |
| IoTHubProtocol | MQTT  AMQP  HTTPS |

# Appendix 3: Fail Responses

|  |  |
| --- | --- |
| HTTP Status | 404 Not Found |
| HTTP Content |  |
| Reason | No Such Object or API Endpoint |

|  |  |
| --- | --- |
| HTTP Status | 401 Unauthorized |
| HTTP Content | {  "Message": "Authorization has been denied for this request."  } |
| Reason | Invalid access token or access token expired |